UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTREAL, QUEBEC H3C 3J7

A NEW APPROACH OF CREW PAIRING PROBLEMS BY COLUMN GENERATION AND APPLICATION TO AIR TRANSPORTS

by

Sylvie Lavoie*
Michel Minoux**
Édouard Odier***

* Air France and Centre de recherche sur les transports of the Université de Montréal.
** LAMSADE, University Paris-IX-Dauphine (France).
*** Head of Operations Research Group, Air France.

Université de Montréal
Centre de recherche sur les transports - Publication #439
November 1985

Dépôt légal - 2e trimestre 1986

REPORT DOCUMENTATION FORM

1. Title	2. Report Date
A New Approach of Crew Pairing Problems by Column Generation and Application to Air Transports.	November 1985
	3. C.R.T. Report No. 439
	4. ISBN

5. Author(s)

Sylvie Lavoie, Michel Minoux, Édouard Odier

6. Sponsoring Organization Name and Address

Air France, FRANCE

7. Abstract

We propose a new approach to crew-pairing problems arising in the context of airline companies. The problem is first formulated as a large scale set covering problem with many columns, each column representing a valid crew-pairing. We then suggest a solution procedure for the continuous relaxation of this large scale problem, based on generalized linear programming, in which the column generation subproblem is shown to be equivalent to a shortest path problem in an associated graph. Computational results obtained on a series of real problems (involving up to 132 flight segments) are reported, confirming both computational efficiency and practical practical applicability of the new approach. Indeed not only were the resulting solutions observed to be integral for all test problems, but savings of about 4 to 5% over the previous approaches were shown to be obtained.

8. Key Words	9. No. of Pages : 27
air transportation, crew-scheduling, column generation	10. Price Code : A02

ABSTRACT

We propose a new approach to crew-pairing problems arising in the context of airline companies. The problem is first formulated as a large scale set covering problem with many columns, each column representing a valid crew-pairing. We then suggest a solution procedure for the continuous relaxation of this large scale problem, based on generalized linear programming, in which the column generation subproblem is shown to be equivalent to a shortest path problem in an associated graph. Computational results obtained on a series of real problems (involving up to 132 flight segments) are reported, confirming both computational efficiency and practical applicability of the new approach. Indeed not only were the resulting solutions observed to be integral for all test problems, but savings of about 4 to 5% over the previous approaches were shown to be obtained.

1. INTRODUCTION

This paper describes a new approach to the well-known, and by now classical, problem of optimally assigning crews to flight services in airline companies, the so-called crew pairing problem (cf for instance AGARD & al. (1967)).

Though the problem may take on several slightly different forms, we will deal here with one of the most frequently encountered variant where each possible feasible crew pairing (= sequence of flight services where a flight service is a sequence of flight segments) is assigned a cost (depending on the various flight services which compose the crew pairing and other parameters such as total flight time etc., see section 2 below) and where it is required to find a set of crew pairings covering all the flight segments at minimum total cost.

Most of the work devoted so far to the crew-pairing problem (CPP) has been carried out in the following direction. Since in practical real-size applications the total number of possible (feasible) crew pairings is enormous, enumeration is impossible, and one must be content with generating once for all a (sufficiently large) number of "good" crew pairings according to appropriate heuristic criteria (typically, for problems involving a few hundreds of flights, a few thousands of crew pairings are considered). The problem is then recognized as a set covering problem which, as shown by many authors, can be efficiently solved using various integer programming techniques such as cutting planes, branch and bound, subgradient optimization and so on (see for instance DELORME (1974), BALAS & HO (1980)).

The new approach presented here, and first presented by Minoux (1984), is based on a generalized linear programming procedure applied to the continuous relaxation of the (large scale) set covering problem

obtained by considering implicitely all the feasible crew pairings. As is well known, the generalized linear programming technique allows solving such large scale (continuous) linear programs to optimality while explicitely considering only a limited number of columns (crew pairings) provided that, at each step of the (revised) simplex algorithm, there exists an efficient algorithm (called the "column generator") for determining the minimum reduced cost column. This is shown to be the case for crew-pairing problems by proving in section 4 that the column generation process can be reduced to shortest path calculations in an associated (acyclic) graph.

The computational results obtained on a series of real test problems, and reported in section 3, fully confirm the various advantages over the previous approaches which could be a priori expected from the new solution procedure :

- improvement in the costs of the solutions obtained (since a much larger set of feasible crew pairings is indeed taken into account) reaching 4 to 5% in real instances over the best previously known solutions;

- higher frequency of integral solutions as compared with optimal continuous solutions of the (restricted) set covering problem obtained when generating, once for all, a subset of "good" crew pairings (indeed for all test problems, the optimal continuous solutions resulting from the column generation procedure were observed to be integral).

Moreover, computing times reported seem to indicate that the new approach could be even more efficient than the previous ones.

2. PROBLEM STATEMENT AND BASIC TERMINOLOGY

The crew pairing problem may be stated as follows.

An airline company has to carry out, within some given time period (usually one or two weeks) a number of flight segments of various origins and destinations. The flight segments are numbered 1, 2, ..., NF. To operate all these flight segments, the company will need a number of crews, and has to assign each crew to a crew-pairing (CP). A crew pairing is a sequence of flight services (with return to the starting base) and a flight service is a sequence of flight segments (which can be performed each after the other). Not any sequence of flight services can be considered as a valid crew pairing and not any sequence of flight segments can be considered as a valid flight service, since they should meet a number of requirements resulting from the current reglementation (e.g. the rest time between two successive flight services within a given crew pairing must be the greater, the larger the corresponding flight time).

We can represent graphically a set of flight services on the time axis, as indicated on Figure 2. For a crew-pairing (such as the one formed by flight services 2, 5, 8 of Fig. 1) we call Stop time in call (STC) the time interval separating two successive flight services in the CP. We call Total absence time (TAT) the total time spent between the departure of the first flight service and the arrival of the last flight service in the CP (for example on Fig. 1, TAT = $t_6 - t_1$). Finally, we call Total flight time (TFT) the sum of the Flight periods (FP) of all the flight services in the CP (for example on Fig. 1, TFT = $(t_2 - t_1) + (t_4 - t_3) + (t_6 - t_5)$), assuming that the Flight period of each flight service is exactly the difference between start time and arrival time of the flight service (in the practical problems which have been dealt with, each flight service is, for its own, composed of 1, 2 or 3 successive flight segments, so the Flight period of a flight service is, in such a situation, less than (arrival time - start time) of the flight service).

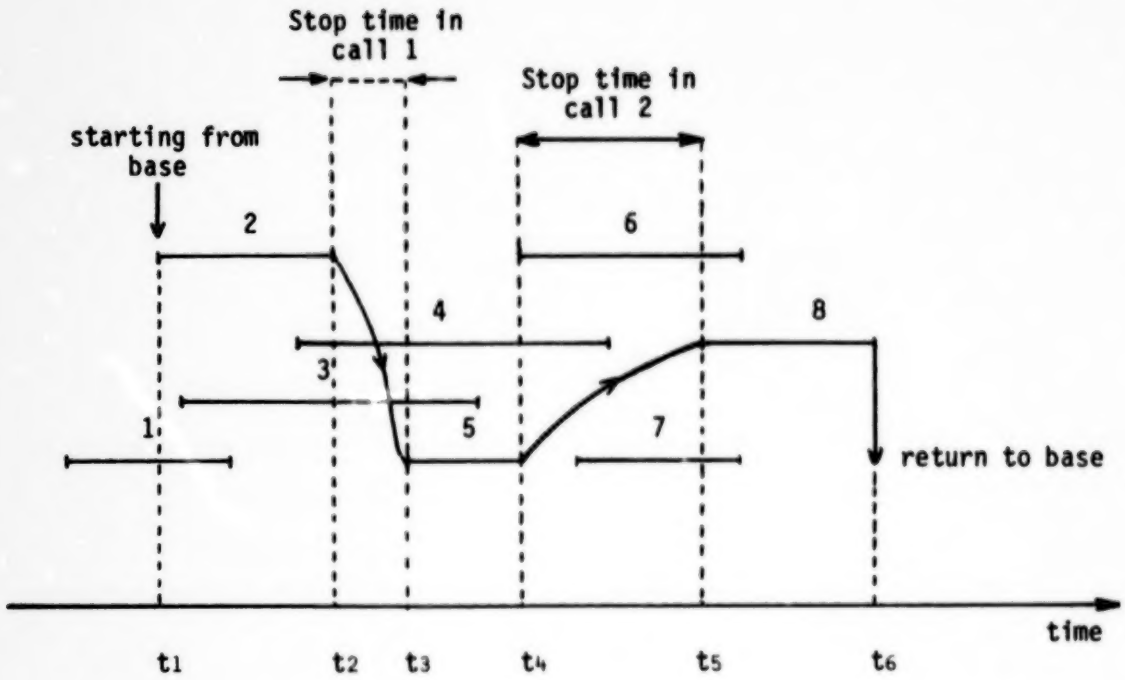


Figure 1 : Representation on the time axis of the set of flight services, and a crew pairing corresponding to the sequence of flight services (2, 5, 8).

3. IDENTIFYING VALID CREW PAIRINGS AS THE SET OF PATHS IN A GRAPH

The first step in building crew pairings is the construction of all valid flight services, a crew pairing being a sequence of flight services. Since a valid flight service is composed of a limited number of flight segments (less than 5 flight segments) satisfying a few additional constraints, we could easily obtain the set of all valid flight services by enumeration of the flight services and verifications of a few constraints.

Now, in order to build a valid CP, a sequence of flight services has to meet a number of complicated requirements, essentially aiming at insuring sufficient STC's between two successive flight services, f_i and f_j, say, within the CP. The difficulty here, is due to the fact that the minimum acceptable value for STC does not only depend on f_i and f_j, but equally on the flight services and/or the STC's performed prior to f_i. Let us illustrate this on a typical, though fancy example.

Suppose that, for flight services of about 5 hours duration, the "standard" minimum STC is 12 hours. If the reglementation included only this "standard" case, then the whole set of valid CP's would be fairly easy to identify. Indeed, starting from a chart such as that of Fig. 1, we could build a graph G :

- the nodes of which would correspond to all flight services;
- the arc set of which would be defined by linking flight service f_i to flight service f_j if the time interval between the two is at least 12 hours.

In this (very simplified) case, the set of valid CP's could be identified with the set of all paths in G.

However, the actual reglementation usually includes much more complicated rules. For instance, it will be admitted that, for three consecutive flight services f_i, f_j, f_k, if the STC between f_i and f_j is somewhat less than the "standard" value (say, 9 hours instead of 12) then this can be compensated for during next call (STC between f_j and f_k) for which a larger value (say, 16 hours) will be required. Figure 2 illustrates a typical set of constraints which are imposed in practice in order to obtain valid crew pairings.

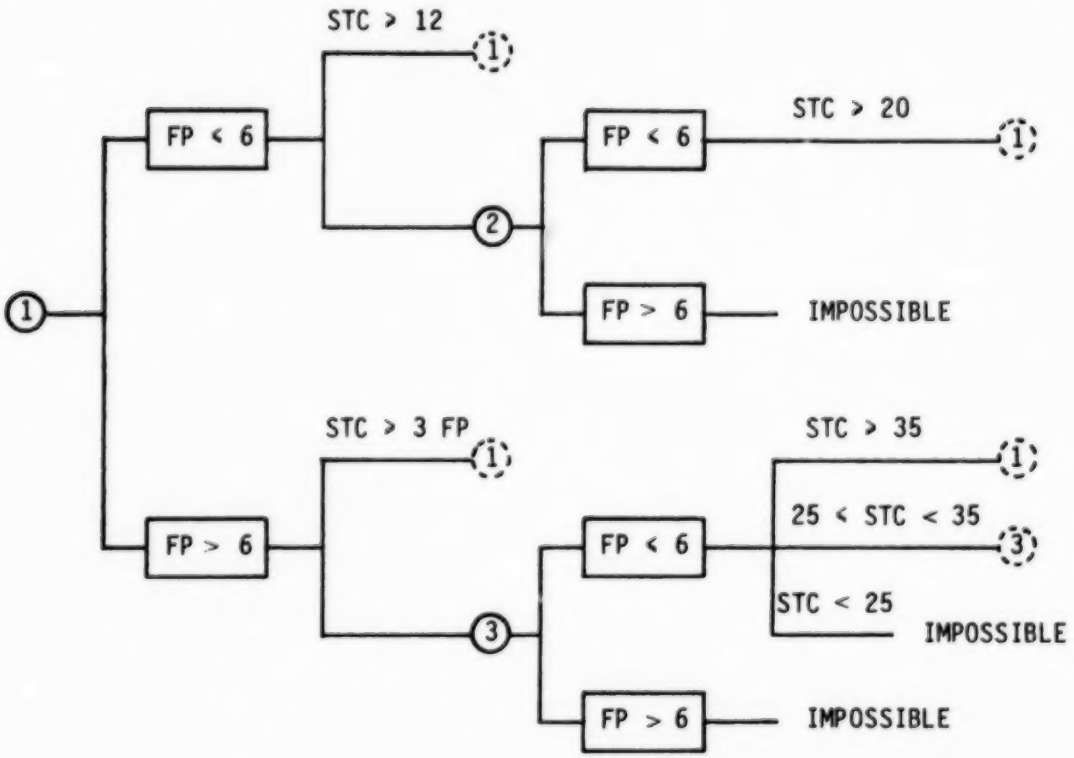


Figure 2 : A typical set of constraints on the STC's (Stop time in call) in order to build valid crew pairings. FP denotes the value of the Flight period and STC the value of the Stop time in call (all are given in hours).

These are seen to be given as a decision tree, the various branches of which correspond to all the possible situations which may occur when a new flight service is added in the course of building a crew pairing. Which branch is followed depends on the actual values taken by the parameters :

FP = value of the Flight period of the flight service under consideration;

STC = value of the Stop time in call between the current flight service and the previous one in the crew pairing being built.

It is observed that the various nodes of the decision tree are given labels ① ② ③ or "impossible" which correspond to various states of the crew pairing being build. Let us define deficiency as the situation in which the actual Stop time in call is less than the standard value (e.g. on Figure 2, STC<12 in case FP < 6 or STC < 3FP in case FP > 6). Then :

- State ① corresponds to the standard case where no deficiency is generated (due to a sufficiently large value of STC).

- State ② corresponds to a deficiency due to a STC less than 12 following a flight service with FP < 6. In that case, the branches of the decision tree following state ② indicate in which way this deficiency can be compensated for by the next flight service. In the example on Figure 2 only two situations may occur depending on whether the next flight service has FP < 6 (in which case a minimum STC of 20 hours will be eventually required resulting in finding again state ①) or FP > 6 (which is ruled out by the "impossible" state).

- State ③ corresponds similarly to a deficiency due to a STC less than 3 FP following a flight service with FP > 6. In that case, the branches of the decision tree following state ③ indicate the various ways in which this deficiency can be compensated

for during the next flight service. For instance, if the next flight service has FP < 6 and is followed by a STC > 35, then state (1) is found again. If the next flight service has FP < 6, it can also be followed by a STC between 25 and 35, in which case we remain in state (3) (thus the crew pairing currently being built may remain in state (3) for several successive flight services). But, starting with state (3): if the next flight service has FP < 6 the eventual STC cannot be less than 25; neither can the next flight service have FP $\geq$ 6 (in both cases the "impossible" state is reached).

It can thus be realized that this concept of a state is quite appropriate for keeping into memory all the information (about the various flight services and STC's preceding the current flight service in the crew pairing) which is relevant to constraint satisfaction. We now show that it can be used to generate a new ("expanded") graph G' in which all the paths are in 1-1 correspondence with the valid crew pairings (i.e. the crew pairings satisfying the constraints specified by the decision tree of the type shown in Figure 2).

The node set in G' is formed by making, for each node in G, as many copies as there are different states in the decision tree (the "impossible" state does not have to be considered). Then, in our example, G' will have 3 times the number of nodes in G : each node in G' corresponds to some flight service together with some status $s \in \{1,2,3\}$. Two nodes in G' are then connected by an arc if the corresponding STC satisfies the constraints specified by the decision tree (see Figure 3).

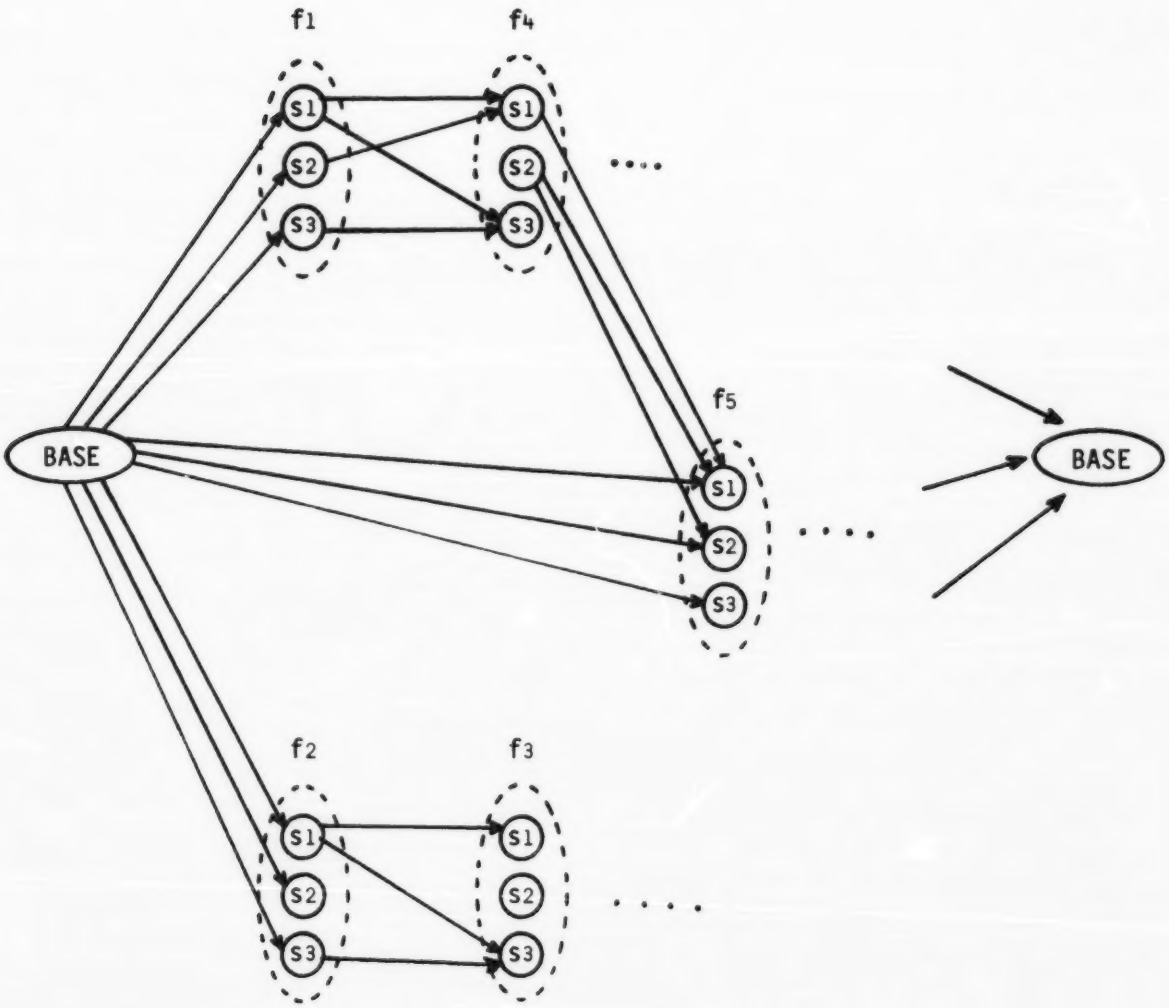
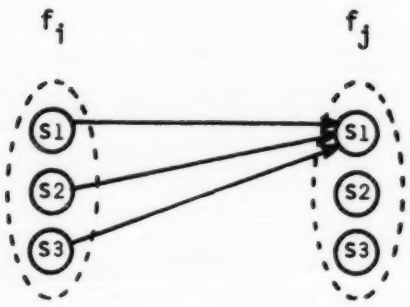


Figure 3 : Structure of the graph G' in which the paths correspond to valid crew pairings.

In order to properly understand how the construction of G' is carried out in the case of the decision tree shown in Figure 2, let us look at a few particular examples.

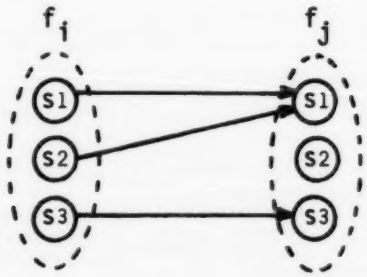
Example 1 :



d_i = 15 (departure) d_j = 56
a_i = 20 (arrival) a_j = 58
FP = 5 FP = 2

In this case, STC = $d_j - a_i = 34$. For such a high value of STC, it can be seen that all the deficiencies which may have been previously generated in building the crew pairing up to f_i, will be compensated for when appending f_j (one always returns in state (1) independently of the state of f_i). In that case, only the three arcs shown on the above figure will have to be considered.

Example 2 :



d_i = 22 d_j = 56
a_i = 26 a_j = 58
FP = 4 FP = 2

In this example f_i has FP = $a_i - d_i = 4$ and STC = $d_j - a_i = 30$. As shown on Figure 2, the STC value is not large enough for all the types of deficiency

to be compensated. Thus if we start from f_i in state (1) or state (2), f_j is reached in state (1) since STC > 12. But if we start from f_i in state (3), f_j is reached in state (3) again since FP < 6 and 25 < STC < 35.

The various examples above give a good flavor of the type of constraints which must be dealt with when coping with practical crew pairing problems in airline companies. However in real situations the decision tree may be much larger than the one shown in Figure 2. For the problems which have been studied at Air France, it was composed of about 25 branches. However, the number of states necessary to describe all possible situations was observed to be not greater than 6; moreover, three among these 6 states corresponded to situations which were almost never encountered in practice, so all our numerical experiments could be carried out with only three states (G' having only three times the number of nodes in G) without significantly affecting the realism of the model. Finally, observe that the generation of the expanded graph G' from the problem data (departure and arrival times of the flights) and the decision tree, can be performed <u>once for all</u> in a <u>preprocessing phase</u> independent from the process of solving the crew pairing problem itself.

4. FORMULATION OF THE CREW PAIRING PROBLEM AS A LARGE SCALE SET COVERING PROBLEM

In view of this, the crew-pairing problem (CPP) can be stated as follows.

(CPP) | Determine a set of CP's such that
- each flight segment belongs to one of the selected CP's;
- the Total cost of the selected CP's is minimum.

The cost of a CP is taken to be a time value, defined as the sum of the Total absence time (TAT) and of a Rest time (RT) :

cost = TAT + RT

where RT is, in general a nonlinear (and discontinuous) function of TAT and of TFT (Total flight time). In practice, it is reasonable to assume that RT is a function of TAT only. Thus RT depends only on the first and last flight services set in the crew pairing.

The (CPP) may then be formulated as a (large scale) set covering problem as follows.

Let K be the total number of valid CP's (= the number of paths in graph G' above). With each CP k ($1 \leq k \leq K$) associate a 0-1 vector R(k) with components $R_i(k)$ (i=1...NF) defined by :

$R_i(k) = 1$ if flight segment i belongs to crew pairing k

$R_i(k) = 0$ otherwise.

With each CP k, also associate its cost c_k, and a 0-1 variable x_k such that :

$x_k = 1$ if the crew pairing k is selected in the solution

$x_k = 0$ otherwise.

The problem is then equivalent to the 0-1 integer linear program (set covering problem).

$$(SCP) \quad \begin{cases} \text{Minimize} \quad \sum_{k=1}^{K} c_k \, x_k \\ \text{s.t.} \\ \sum_{k=1}^{K} R(k) \cdot x_k \geq \mathbb{1} \\ x_k = 0 \text{ or } 1 \quad (k=1,\ldots,K) \end{cases}$$

($\mathbb{1}$ is the vector of dimension NF with all components equal to 1).

(SCP) contains a limited number of constraints (equal to the number of flight segments NF) but a usually huge number of variables.

5. SOLVING THE LARGE SCALE SET COVERING PROBLEM BY COLUMN GENERATION

The classical approach to the crew pairing problem consists in solving a restricted problem, obtained by generating, once for all, according to heuristic criteria, a subset of "good crew pairings". (Typically, for NF $\approx$ 100, 1000 to 2000 CP's are generated). The restricted problem is then solved as an integer program, e.g. by cutting-plane techniques (cf DELORME 1974 for instance).

The main drawback in this approach is that one can never be sure to have, indeed, generated the best possible CP's since there is no kind of interaction between the solution procedure, and the column generation procedure. To get rid of this drawback, we propose here a new approach based on a column generation scheme, ensuring a constant two-way interaction between the solution procedure and the column generation procedure. The idea is here that it is more important, at least as a first step, not to exclude a priori any valid CP (by implicitely working on the entire set of CP's) than to take account of the integrality constraints on the variables. Going even further in this direction, it can be shown that, at least for some specific subclass of problems of the crew-pairing type, the column generation method will produce optimal integer solutions, while, in the previous approaches, the optimal continuous solution to the restricted problem is in general noninteger (see section 6.3).

Let us restrict here to show how (SCP) can be solved (as a continuous linear program) by generalized linear programming. At the current step, t, suppose that a subset $J_t \subset J = \{1,2,\ldots,K\}$ of valid CP's has been generated. Solving the problem restricted to the columns in J_t, i.e. :

$$(PP)\quad \begin{cases} \text{Minimize} \quad \sum_{k \in J_t} c_k \, x_k \\ \text{s.t.} \\ \sum_{k \in J_t} R(k) \cdot x_k \geq \mathbf{1} \\ 0 \leq x_k \leq 1 \qquad (k \in J_t) \end{cases}$$

produces a set of simplex multipliers (optimal dual variables) Π_i $(i=1,\ldots NF)$. Thus, with each flight segment i, a multiplier Π_i is associated. The column generation subproblem then consists in looking for a minimum reduced cost crew pairing, among all possible CP's i.e. :

$$\bar{c}_{k_o} = c_{k_o} - \sum_{i=1}^{NF} \Pi_i \, R_i(k_o) = \min_{k=1\ldots K} \{c_k - \sum_{k=1}^{NF} \Pi_i \, R_i(k)\}$$

Now, this k_o may be obtained without need of enumerating all the columns (all the CP's) by noticing that it reduces to a shortest path problem in the graph G' (defined above) where each node corresponding to a flight service f_i is assigned a length Π_i (the graph G' being acyclic, there is no problem with negative lengths) . The column generation subproblem may thus be solved via a simple labelling procedure in the acyclic graph G' such as the one suggested by DIJKSTRA (1959) and which has been used to carry out the numerical experiments reported in the following section. Since the cost of the crew pairing (TAT + RT) was assumed to depend only on the first and last flight services (see section 4), the computation of the minimum reduced cost $\bar{c}_{k_o}$ can be decomposed into several labelling steps, one for each flight service f_i taken as the origin. For any flight service f_j, the minimum reduced cost of the crew pairing starting with f_i and ending with f_j – if any – is obviously obtained by adding to the label of f_j the cost (TAT + RT) which is a constant for the pair (f_i, f_j).

6. COMPUTATIONAL RESULTS

The column generation procedure described in section 5 above has been implemented and applied to a number of real test problems concerning commercial flying agents (stewards and hostesses) on Air France long-distance network. Experimentation has been carried out on a IBM 3081 computer, using the MPSX linear programming package. (However, only a limited part of MPSX has been used, in particular, the integer programming options were left aside.)

The computational results are divided into three parts, subsumed in Tables I to III.

Table I concerns a single (real) problem and provides a series of results all obtained on this problem, in order to study the effect of varying the parameter fixing the maximum allowable Total absence time (TAT).

Table II provides a series of results all obtained on a single (real) problem and aimed at exploring the sensitivity of the algorithm to various starting solutions (i.e. various distinct ways of initializing the restricted problem).

Finally table III displays results obtained on a series of various distinct test problems, either real problems or problems derived from a real problem by performing minor changes such as : deleting a few flight services, changing arrival or departure times of some flights, etc.

On each of the three Tables, the following informations will be found :

1) Problem identificator;

2) The size of the problem : number of flight segments;

3) The size (number of nodes and arcs) of the underlying graph G' (the paths of G' correspond to valid crew-pairings);

4) TAT_{max}, the maximum Total absence time allowed for a crew pairing;

5) z^0, the initial value of the objective function corresponding to the starting solution (i.e. forming the starting restricted problem);

6) z^*, the value of the continuous optimum solution to the large scale set covering problem;

7) The total number of columns generated during the column generation process. It should be pointed out here that the total number of columns added to the restricted linear program is twice the number of crew pairings actually generated by means of the "column generator" (shortest path). This is because the practical problems dealt with concerned a typical time period of two consecutive weeks and, for each crew pairing provided by the "column generator", we automatically added the mirror image of this crew pairing relative to the middle of the two-week period. (By "mirror image" it is meant that the flight services belonging to week 1 became those relative to week 2 and vice-versa). This way of implementing column generation has been chosen in order to make the best possible use of the informations obtained each time the column generator was called for, and because, intuitively if a crew pairing is good according to the reduced cost criterion, then its "mirror image" should be good too.

8) Indication concerning the integrality/nonintegrality of the solution obtained.

9) Computation time. It should be noted that the indicated values do not include the (fixed) precomputation time needed for building the underlying graph G'. (This precomputation time is about 10 seconds for all the problems treated).

TABLE I

Problem number	Number of flight segments	Number of flight services	Underlying graph G'		TAT_{max} (hours)	Z^o	Z^*	Total number of generated columns	Number of calls to generator	Integrality of the solution	Computing times (sec.)
			Number of nodes	Number of arcs							IBM 3081
1	132	248	768	6204	337.50	3823.78	3599.82	416	6	YES	56
1	132	248	768	6204	337.50 (14j. + Δ)	5334.84	3599.82	521	6	YES	66
1	132	248	768	6204	313.50	3823.78	3628.92	353	8	YES	67
1	132	248	768	6204	313.50 (13j. + Δ)	5334.84	3628.92	407	7	YES	63
1	132	248	768	6204	288.50	3823.78	3654.68	333	10	YES	62
1	132	248	768	6204	288.50 (12j. + Δ)	5334.84	3654.68	405	8	YES	59
1	132	248	768	6204	265.50	3823.78	3823.78	195	5	YES	30
1	132	248	768	6204	265.50 (11j. + Δ)	5334.84	3823.78	234	6	YES	28

Note : Δ = briefing + Debriefing = 1.00 + 0.50 = 1.50

TABLE II

Problem number	Number of flight segments	Number of flight services	Underlying graph G'		TAT_{max} (hours)	Z^0	Z^*	Total number of generated columns	Number of calls to generator	Integrality of the solution	Computing times (sec.) IBM 3081
			Number of nodes	Number of arcs							
1	132	248	768	6204	289.50	3823.78	3654.68	307	10	YES	79
1	132	248	768	6204	289.50	5334.84	3654.68	333	8	YES	62
1	132	248	768	6204	289.50	7373.34	3654.68	319	9	YES	71
1	132	248	768	6204	289.50	7272.66	3654.68	405	7	YES	59
1	132	248	768	6204	289.50	8710.02	3654.68	299	5	YES	43
1	132	248	768	6204	289.50	9770.84	3654.68	398	6	YES	52

TABLE III

Problem number	Number of flight segments	Number of flight services	Underlying graph G'		TAT_{max} (hours)	Z^0	Z^*	Total number of generated columns	Number of calls to generator	Integrality of the solution	Computing times (sec.) IBM 3081
			Number of nodes	Number of arcs							
1	132	248	768	6204	289.50	5334.84	3654.68	333	6	YES	62
2	20	38	122	322	289.50	2478.00	479.82	4	2	YES	4
3	132	244	764	6330	289.50	11312.89	5469.32	316	7	YES	52
4	132	246	774	6384	289.50	11992.30	3656.34	463	6	YES	54

6.1 Behaviour of the solution costs as the maximum allowed Total absence time varies

Table I displays the results obtained on a single (real) problem for four distinct values of parameter TAT_{max} which represents an upper bound on the Total absence time allowed in a crew pairing. For each of these values of TAT_{max} (337.50, 313.50, 288.50, 265.50) two starting solutions have been considered in order to demonstrate independence of the results on the way the algorithm is initialized.

From these results, it can be seen that an increase of the parameter TAT_{max} induces a significant improvement in the final optimal value of the objective function (see Figure 4). The best solution previously known for this problem (and which was obtained by hand) corresponded to the optimal value obtained for TAT_{max} = 265.50. Allowing the parameter TAT_{max} to take values equal to 289.50 (12 days + briefing-debriefing) and 313.50 (13 days + briefing-debriefing) improvements of 4.4% and 5.1% were obtained respec-

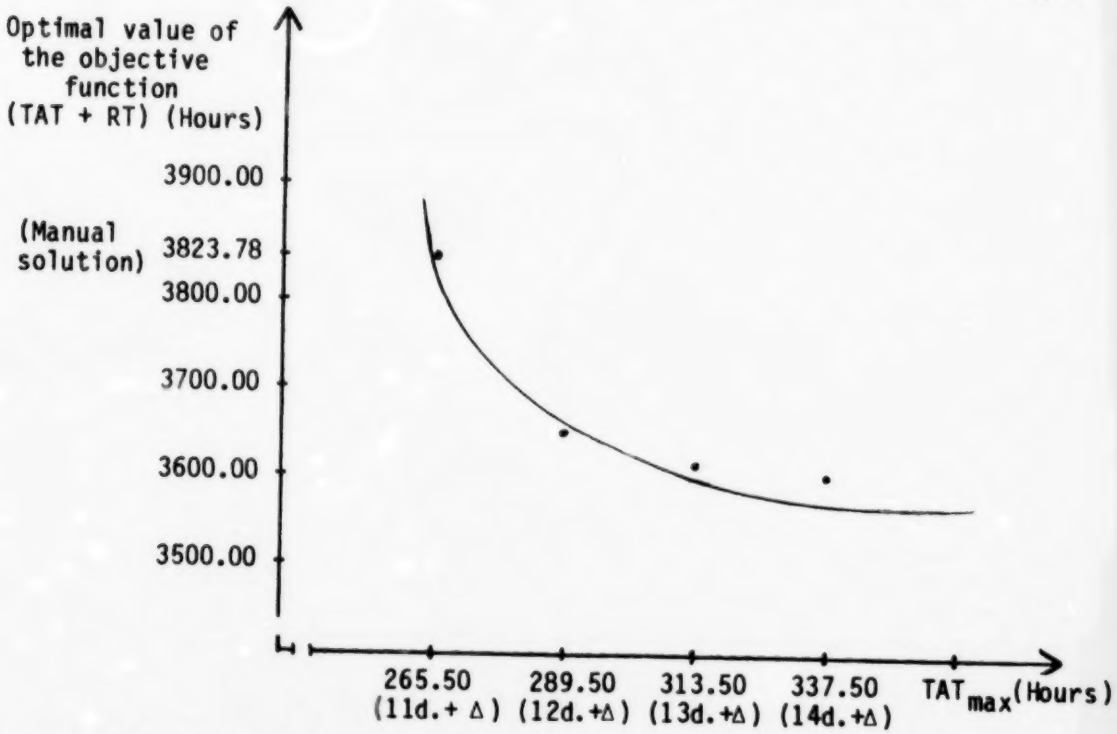


Figure 4 : The variation of optimum objective function value versus maximum allowed Total absence time (TAT_{max}).

tively which is very significant. Moreover, it can be observed that the computing times are about the same order of magnitude for all the examples treated, and that all the optimal solutions obtained are integers.

As can also be seen the model appears to be quite well-suited to carry out sensitivity analyses of the above type.

6.2 Behaviour of the model with respect to the choice of a starting solution

Table II displays the results obtained on a single (real) problem with six distinct starting solutions. For all six series of tests, the value of the parameter TAT_{max} has been set to 289.50 (11 days + briefing-debriefing). The following observations can be drawn. First it is noticed that the computing times as well as the total numbers of generated columns are all very comparable, confirming a quite stable behaviour of the solution algorithm. In particular, it is seen that the way the algorithm converges to the optimal solution is, to a great extent, independent of the quality of the starting solution; thus the algorithm does not require prior knowledge of a very good starting solution to achieve good performance. (Actually, in some of the experiments shown in Table II, illegal crew pairings were even introduced - with a sufficiently large associated cost - into the starting solution, without deteriorating the performances).

6.3 Behaviour of the model on a variety of problems

Table III shows the results obtained on a series of four distinct problems. For all these examples, the value of parameter TAT_{max} (the maximum allowed Total absence time in a crew-pairing) has been set to 289.50, and the starting solution was chosen by hand, in an arbitrary way. Since setting up a complete set of real data requires considerable work, the problems considered were all derived from a single (real) problem (the same as the one considered in sections 6.1 and 6.2 by performing minor changes. To be more specific :

- Problem 2 has been deduced from Problem 1 by discarding a few flight segments;
- Problem 3 has been deduced form Problem 1 deleting a few flight services;
- Problem 4 has been deduced form Problem 1 by performing a few modifications in the departure and/or arrival times of a few flights.

Here again we observe, as in section 6.2 above, a good stability of the model in particular concerning computing times and total number of generated columns. Also, a very interesting feature of the algorithm is that, at least in the computational experiments reported here, the optimal solutions obtained all have integral (0-1) components, thus avoiding recourse to some sophisticated integer programming schemes (such as Branch and Bound, or cutting planes).

Thus the new solution method enjoys the nice property of providing apparently much more frequent integer solutions than the previous approaches. A good intuitive explanation of this phenomenon is the following. Consider the simplified version of the problem already alluded to at the beginning of section 3, where all the valid crew pairings are in 1-1 correspondence with the set of paths of some graph G. Also, consider the case where each flight service is composed of one flight segment and where the cost criterion is reduced to a linear cost.

In this case, using an arc-node formulation, we could write the problem as a network flow problem. If this problem is solved via a standard minimum cost flow algorithm (such as the ouf of kilter method, see FORD & FULKERSON, 1962) we know that an integral solution will result (this is a consequence of the so-called integrality property of network flow problems). However, another way of looking at the problem is to use an arc-path forlumation (based on the decomposition of flows into paths) in which case we find again exactly the large scale set covering problem (SCP) introduced in section 4. Since the column generation algorithm implicitely

considers all the columns (all the possible ways of decomposing the flow into paths) it will produce an optimal solution to the network flow problem which, as already pointed out, should be integral (we do not consider here the possible, though unfrequent, cases of degeneracy where the column generation algorithm could produce a fractional optimal solution in spite of the fact that there exists an integral optimal solution).

Now suppose we try and solve the problem using the traditional approach of crew pairing problems, namely : a set of paths is generated, once for all, according to good heuristic criteria (e.g. retaining only low cost crew-pairings corresponding to low cost paths in the graph) thus forming a restricted set covering problem. Then, if the restricted problem is solved as an ordinary linear program, the probability of having generated all paths involved in an optimal solution to the original (network flow) problem is very low, hence it is rather unlikely that we obtain an optimal (integer) solution. Considering this class of problems which can be reduced to network flow problems thus provides some support to the empirical observation that the column generation technique, by considering (implicitely) all the possible valid crew pairings, indeed greatly increases the probability of getting integral optimal solutions.

ACKNOWLEDGEMENTS

Special thanks are due to Jacques Desrosiers and François Soumis, from the Centre de recherche sur les transports of the Université de Montréal, for fruitful discussions on the implementations of the column generation technique. Françoise Lascaux from the Operations Research Department at Air France is also gratefully acknowledged for her kind and professional help.

APPENDIX I - Notation

The following symbols are used in this paper :

CP : crew-pairing

(CPP) : crew pairing problem

STC : Stop time in call separating two successive flight services in the CP

TAT : Total absence time for a CP

FP : Flight period of a flight service

TFT : Total flight time of a CP

RT : function evaluating the Rest time of a CP

(SCP) : set covering problem

REFERENCES

AGARD, J., ARABEYRE, J.P., VAUTIER, J. (1967) : "Génération automatique de rotations d'équipages", RAIRO 6, 107-117.

BALAS, E., HO, A. (1980), "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization = A Computational Study", Mathematical Programming Study 12, 37-60.

DANTZIG, G.B. (1963), "Linear Programming and Extensions", Princeton University Press.

DELORME, J. (1974), "Contribution à la résolution du problème de recouvrement : méthodes de troncatures", Thèse, Docteur Ingénieur, Université Paris VI, Paris.

DESROSIERS, J., SOUMIS, F. and DESROCHERS, M. (1984), "Routing with Time Windows by Column Generation", Networks 14, 545-565.

DIJKSTRA, E.W. (1959), "Numerische mathematik", Vol. 1, 269-271.

FORD, L.R., FULKERSON, D.R. (1962), "Flows in Networks", Princeton University Press.

GILMORE, P.C., GOMORY, R.E. (1961), "A Linear Programming Approach to the Cutting-Stock Problem", Operations Research 9, 849-869.

GONDRAN, M., MINOUX, M. (1979), "Graphes et algorithmes", Eyrolles, Paris.

LAVOIE, S. (1984), "Mise en oeuvre d'un algorithme de génération de colonnes pour le problème de rotation des équipages", Final report of the crew-pairing project carried out at AIR FRANCE.

MINOUX, M. (1975), "Plus courts chemins avec contraintes : algorithmes et applications", Annales des Télécommunications (France), tome 30, n° 11-12, 383-394.

MINOUX, M. (1983), "Programmation mathématique : théorie et algorithmes", tomes I et II, Dunod, Paris.

MINOUX, M. (1984), "Column Generation Techniques in Combinatorial Optimization, a New Application to Crew-Pairing Problems", Proceedings XXIVth AGIFORS Symposium - Strasbourg - France (Sept. 1984).

ACKNOWLEDGEMENT

We would like to thank especially François Soumis and Jacques Desrosiers from the Centre de recherche sur les transports of the Université de Montréal for their help from both the theoretical and technical point of view.